(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 September 2013 (06.09.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/128331 A1

(51) International Patent Classification:
G06F 12/00 (2006.01) *G06F 17/21* (2006.01)

(21) International Application Number:
PCT/IB2013/051343

(22) International Filing Date:
19 February 2013 (19.02.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/407,499 28 February 2012 (28.02.2012) US

(71) Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).

(71) Applicants *(for MG only)*: **IBM UNITED KINGDOM LIMITED** [GB/GB]; PO Box 41, North Harbour, Portsmouth, Hampshire PO6 3AU (GB). **IBM JAPAN LIMITED** [JP/JP]; 19-21 Nihonbashi, Hakozaki-cho, Chuo-ku, Tokyo 103-8510 (JP).

(72) Inventors: **WENDT, David, Mark**; IBM Corporation, M/P CPFA/502/J222, P O Box 12195, 5YIA/503/M126, 3039 Cornwallis Road, Research Triangle Park, NC 27709-2195 (US). **KUBIK, Joseph**; IBM Corporation, M/P CPFA/503/M113, P O Box 12195, 5YIA/503/M126, 3039 Cornwallis Road, Research Triangle Park, NC 27709-2195 (US). **BAREFOOT, Christopher, Boyd**; IBM Corporation, Intellectual Property Law (J46/G4), 555 Bailey Avenue, San Jose, CA 95141-1003 (US). **RICH, Timothy, Slade**; IBM Corporation, M/P J303/B502, P O Box 12195, 5YIA/503/M126, 3039 Cornwallis Road, Research Triangle Park, NC 27709-2195 (US).

(74) Agent: **STRETTON, Peter**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester, Hampshire SO21 2JN (GB).

[Continued on next page]

(54) Title: ON-DEMAND FILE SYNCHRONIZATION

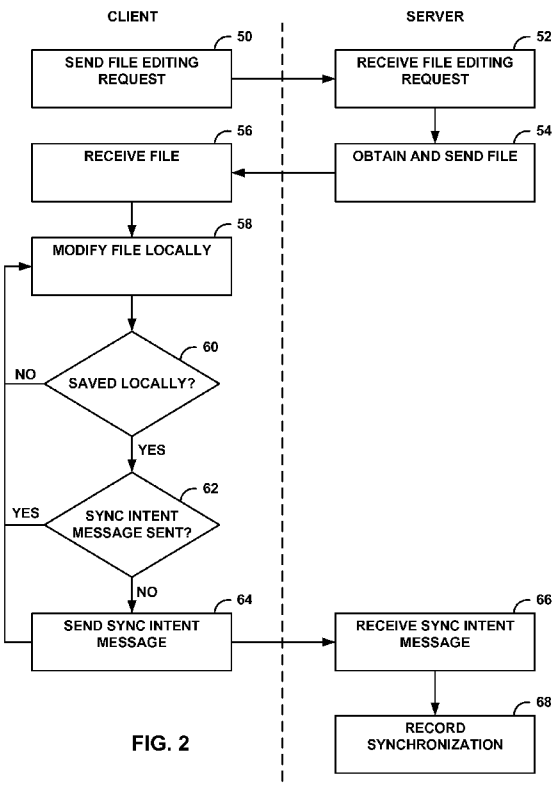

FIG. 2

(57) Abstract: Techniques are described for managing access and synchronization of one or more files of a document management system stored locally at a client device. The techniques may include receiving, by a document management system executing on a server device, a file update notification message from a first client device to notify the document management system that a file was modified by the first client device. The file update notification message may include an indication of the modified file on the first client device without including the modified file. The document management system may receive a file editing request from a second client device to request the file. In response, the document management system may send a file upload request to the first client device, receive the modified file from the first client device, and send the modified file to the second client device.

WO 2013/128331 A1

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

ON-DEMAND FILE SYNCHRONIZATION

BACKGROUND

[0001] This disclosure relates to software systems, and more specifically, to document management systems.

[0002] Document management systems may be used to maintain and organize electronic documents. Such document management systems are typically used in a collaborative file sharing context, thus enabling multiple users to access and edit a single electronic document using multiple computing devices connected using a computing network. Document management systems may support many, e.g., hundreds or thousands, of concurrent users. For instance, document management systems may be used in conjunction with sophisticated, large-scale enterprise software systems, such as inventory management systems, budget planning systems, order management systems, sales force management systems, business intelligence tools, enterprise reporting tools, project and resource management systems, and other enterprise software systems.

[0003] Typically, files within a document management system are stored at a central location, such as a server. A user computing device (e.g., a client device) may transmit a request to the server to access a file for editing. In response, the server typically sends a copy of the file that is stored at the central server to the client device. In addition, the server may associate a "lock" for the file with the requesting client device, thereby ensuring that editing requests originating from other client devices are rejected until the editing client "unlocks" the file. Files are typically modified locally by the client device and uploaded back to the central server that maintains the current version of the file.

BRIEF SUMMARY

[0004] Techniques are described herein in which a document management system manages access to a current version of a file that is stored locally on a client device. Conventionally, current versions of files within a document management system are stored at a central

[0005] location, such as a server. As an example, a user of a client device may "lock" a file for editing and download the file to the client device, thereby effectively "checking out" the file Upon completion of the editing process, a user of the client device "checks in" the file by uploading the modified file to the server and releasing the lock. As another example, client-side software may monitor the file for changes and automatically upload the file or the incremental changes to the server when the changes occur. As such, the server typically maintains the current version of the file when the file is not being edited.

[0006] In contrast, techniques are described herein in which the current version of a file associated with a document management system is stored locally at a client device. That is, the current version of the file is stored by the client device that most recently edited the file even though the user has finished editing the file and has effectively checked the file back into the document management system. Rather than transmit an updated file to the server upon completion of the editing process, or each time the file is saved, the techniques allow the updated file to be transmitted on demand when requested by other users for subsequent editing. As such, techniques described herein may reduce the amount of network communications required to manage access to a current version of a file in a document management system. This may be advantageous when, for example, network bandwidth is low, file sizes are large, or the system includes a large number of client devices.

[0007] In one example, a method includes receiving, by a document management system executing on one or more processors of a server device, a file update notification message from a first client device to notify the document management system that a file was modified by the first client device. The file update notification message may include an indication of the modified file on the first client device without including the modified file. The method further includes receiving, by the document management system, a file editing request from a second client device to request the file from the document management system, and responsive to receiving the file editing request from the second client device, sending, by the document management system, a file upload request to the first client device to request the modified file from the first client device. The method further includes receiving, by the document management system, the modified file from the first client device, and sending, by the document management system, the modified file to the second client device.

[0008] In another example, a computer-readable storage medium is encoded with instructions that, when executed, cause one or more processors of a server device to receive, by a document management system executing on the one or more processors, a file update notification message from a first client device to notify the document management system that a file was modified by the first client device. The file update notification message may include an indication of the modified file on the first client device without including the modified file. The computer-readable storage medium is further encoded with instructions that, when executed, cause the one or more processors of the server device to receive a file editing request from a second client device to request the file from the document management system, and responsive to receiving the file editing request from the second client device, send a file upload request to the first client device to request the modified file from the first client device. The computer-readable storage medium is further encoded with instructions that, when executed, cause the one or more processors of the server device to receive the modified file from the first client device, and send the modified file to the second client device.

[0009] In another example, a server device includes one or more processors, and a document management system executable by the one or more processors to receive a file update notification message from a first client device to notify the document management system that a file was modified by the first client device. The file update notification message may include an indication of the modified file on the first client device without including the modified file. The document management system is further executable by the one or more processors to receive a file editing request from a second client device to request the file from the document management system, and responsive to receiving the file editing request from the second client device, send a file upload request to the first client device to request the modified file from the first client device. The document management system is further executable to receive the modified file from the first client device and send the modified file to the second client device.

[0010] In another example, a method includes modifying a file by a client device, storing the modified file at a computer-readable storage media of the client device, and sending, by the client device, a file update notification message to a document management system

executing on one or more processors of a server device to notify the document management system that the client device modified the file. The file update notification message may include an indication of the modified file stored at the computer-readable storage media of the client device without including the modified file.

[0011] In another example, a computer-readable storage medium is encoded with instructions that, when executed, cause one or more processors of a client device to modifying a file, storing the modified file at the computer-readable storage medium, and send a file update notification message to a document management system executing on one or more processors of a server device to notify the document management system that the client device modified the file. The file update notification message may include an indication of the modified file stored at the computer-readable storage media of the client device without including the modified file.

[0012] In another example, a client device includes one or more processors, one or more applications executable by the one or more processors to modify a file, and a computer-readable storage medium operable to store the modified file. The client device further includes a file system monitor executable by the one or more processors to send a file update notification message to a document management system executing on one or more processors of a server device to notify the document management system that the client device modified the file. The file update notification message may include an indication of the modified file stored at the computer-readable storage medium of the client device without including the modified file.

[0013] The details of one or more aspects of this disclosure are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the disclosure will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0014] FIG. 1 is a block diagram illustrating an example communication system for synchronizing files of a document management system, in accordance with one or more aspects of this disclosure.

[0015] FIG. 2 is a flow diagram illustrating an example operation of a client device of the communication system of FIG. 1 in further detail.

[0016] FIG. 3 is a flow diagram illustrating an example operation of a server device of the communication system of FIG. 1 in further detail.

DETAILED DESCRIPTION

[0017] Techniques are described herein in which a document management system manages synchronization and access to a current version of a file that is stored locally on a client device. As described herein, a first client device communicates with the document management system to lock a file for editing and download the file from a server. The first client device stores the file locally and modifies the file at the request of the user. However, upon the file being saved or released by the user upon completion of the editing process, the first client sends a synchronization intent message to the server rather than transmit the modified file to the server. The synchronization intent message notifies the document management system executing on the server that the file has been locally modified and the user has taken an action indicating an intent to synchronize the file with the server.

[0018] In response to receiving a subsequent request to edit the file from a second client device, the server determines whether the first client device has unlocked the file for editing. When the server determines that the file is locked, the server may transmit an unlock request message to the first client device to request the first client device to release the lock. In some examples, as when the user is not finished editing the file, the user may provide an input indicating that the first client device is not to release the lock. In response, the first client device may transmit an unlock rejection message to the server device. When the first client device does not relinquish the lock, the server may reject the file editing request received from the second client device.

[0019] In other examples, in response to an unlock request message or at the request of the user upon completion of the editing process, the first client device may transmit an unlock confirmation message to the server to release the lock and effectively check the file back into the document management system. Rather than transmit the updated file to the server upon checking in the file, the first client device maintains the updated version of the file locally. When the server determines that the file is unlocked, the server sends a file upload request to the first client device. In turn, the first client device transmits the updated version of the file to the server. The server then sends the updated file to the second client device for editing.

[0020] The techniques described herein may reduce the amount of network communications required to manage access to and synchronization of a current version of a file in a document management system. For instance, the synchronization intent message may include an indication of the modified document without including the modified file itself. As such, the amount of network communications required to notify the server of the updated file may be reduced. Moreover, the techniques enable a current version of a file to be stored locally at a client device and transmitted to the server on demand, thereby possibly further reducing the amount of network communications required for document synchronization.

[0021] FIG. 1 is a block diagram illustrating an example communication system 2 for synchronizing one or more files of a document management system 40. Document management system 40 may be any software application that manages synchronization of one or more files among client devices. Document management system 40 enables multiple clients to work collaboratively on an electronic file. For example, document management system 40 may manage synchronization of a file among client devices by controlling each client's access to edit a current version of the file. As such, document management system 40 prevents concurrent editing of a file that may result in one client overwriting changes made to the file by another client.

[0022] Document management system 40 may control access to edit a file using a file locking mechanism. As an example, document management system 40 may receive a request from a client to edit a file. In response, document management system 40 may determine whether the file is currently "locked" by another client device. If the file is

locked, document management system 40 may reject the editing request until the file is "unlocked" by the client device associated with the lock. As such, document management system 40 ensures that only one client device edits the file at any given time, thereby preventing possible conflicting modifications to the file that may result in one client overwriting changes made by another client.

[0023] In the example of FIG. 1, document management system 40 interacts with client devices 4A and 4B (collectively "client devices 4") to manage access to and synchronization of file 22. Although described with respect to client device 4A, client devices 4A and 4B include substantially similar components to perform substantially similar functionality. In addition, although described for exemplary purposes with respect to two client devices, it should be noted that the techniques described herein may be applied with respect to many client devices, e.g., hundreds, thousands, or more client devices. Similarly, although illustrated with respect to server device 38, document management system 40 may be part of, or distributed among, multiple server devices, such as server device cluster 46 illustrated in FIG. 1.

[0024] Client devices 4, in some examples, may include or be part of a portable computing device (e.g., a mobile phone, netbook, laptop, personal digital assistant (PDA), tablet device, and the like), desktop computer, or server. Client devices 4A and 4B may be the same or different type of device. As an example, client device 4A may be a laptop computer and client device 4B may be a mobile phone. As another example, client device 4A and client device 4B may each be desktop computers.

[0025] In the example of FIG. 1, client device 4A executes application 6 to enable a user to perform one or more tasks. For example, application 6 may include one or more word processing applications, spreadsheet applications, slide presentation applications, and the like. Application 6 may enable a user to retrieve and update a file, such as a text file associated with a word processing application. In some examples, application 6 may include or be part of a suite of tools that may enable a user to contribute, retrieve, analyze, and visualize information. For instance, application 6 may be part of a large-scale enterprise software system, such as inventory management systems, budget planning systems, order

management systems, or other types of enterprise software systems. Although discussed herein with respect to a single application, techniques of this disclosure are applicable to multiple applications 6.

[0026] As illustrated in FIG. 1, client device 4A includes operating system 16 and file system 20. Operating system 16 manages hardware resources (e.g., processors, memory, peripheral devices such as a keyboard or mouse) and software components of client device 4A. Operating system 16 provides an interface between such hardware and software components, thereby providing an operating environment for execution of application processes (e.g., application 6, document management agent 9, file system monitor 8, and the like). For instance, operating system 16 may expose one or more application programming interfaces (API) to enable application 6 and file system monitor 8 to interact with hardware resources and software components executing on client device 4A. Such APIs may include, among others, inter-process communication mechanisms and system calls.

[0027] File system 20 may be considered a component of operating system 16 that allocates and organizes storage space for data on an underlying storage media, such as random access memory (RAM), read only memory (ROM), flash memory, a hard disk, or other computer-readable storage media. File system 20 may organize data using a folder or directory structure. File system 20 may be considered an abstraction to enable operating system 16 to store, retrieve, and update data. As such, file system 20 provides an interface between operating system 16 and the underlying physical storage media upon which data such as file 22 is stored. Examples of file system 20 include, but are not limited to, File Allocation Table (FAT), Fast File System (FFS), and New Technology File System (NTSF).

[0028] In the example of FIG. 1, application 6 issues system calls 10 and 12 to cause operating system 16 to interact with file system 20. As an example, system calls 10 may include file management commands to cause file system 20 to save, write, or otherwise modify file 22. System calls 12 may include file management commands to enable application 6 to retrieve information from file 22, such as file open, file read, and other similar file system management commands. As such, operating system 16 provides an interface between file system 20 and applications executing on client device 4A (e.g.,

application 6, document management agent 9, file monitor system 8, etc.) to enable such applications to interact with file 22 stored on an underlying physical storage media of client device 4A.

[0029] As described herein, document management system 40 performs operations to manage access and synchronization of file 22 among client devices 4 to enable client devices 4 to collaboratively edit file 22. As illustrated in FIG. 1, document management system 40 receives editing request 24 from client device 4A. Client device 4A may connect to server 38 using a communications network, such as a local-area network (LAN), a wide-area network such as the Internet, an enterprise network, a telephone network such as a cellular telephone network, or one or more other types of networks.

[0030] Editing request 24 includes an indication of a file for which client device 4A requests access to edit. In this example, editing request 24 includes an indication of file 22, such as a file name, file version, or other unique identifier of file 22. In some examples, editing request 24 may include an identifier of client device 4A, a user of client device 4A, or both. For instance, editing request 24 may include a unique username of a user of client device 4A, a media access control (MAC) address of a network interface of client device 4A, an Internet Protocol (IP) address and port number, or other identifiers to facilitate communications and identification of client device 4A. As another example, client device 4A may "login" to document management system 40 prior to sending editing request 24. In such an example, when logging in, client device 4A may provide identification information of client device 4A, a user of client device 4A, or both, to enable document management system 40 to identify client device 4A as the requesting client.

[0031] In response to receiving editing request 24, document management system 40 accesses database 42 to identify a storage location of a current version of file 22 and whether file 22 is currently locked for editing by a client device. Examples of database 42 include relational databases, multi-dimensional databases, hierarchical databases, object-oriented databases, or one or more other types of databases. Database 42 may be part of server device 38, one or more server devices of server device cluster 46, or distributed among one or more of server device 38 and server device cluster 46. In general, database 42 may

include any organizational structure that enables document management system 40 to associate a current version of a file with a storage location of the file. That is, document management system 40 accesses database 42 to identify the device (e.g., a client device such as one or more of client devices 4, server device 38, one or more servers of server cluster 46, etc.) that stores the current version of file 22 (i.e., the most recently modified version of file 22), and whether the file is currently locked for editing by a client device.

[0032] For instance, after receiving editing request 24, document management system 40 may access database 42 and determine that file 22 is locked for editing (i.e., checked out) by another client device, such as client device 4B. In such an example, document management system 40 may reject editing request 24, thereby preventing client device 4A from editing file 22. In the example of FIG. 1, document management system 40 accesses database 42 to determine that file 22 is unlocked and that the most recently modified version of file 22 is stored in file repository 44 of server device 38. As such, document management system 40 retrieves file 22 from file repository 44 and transmits file 22 to client device 4A. File repository 44 may include any combination of file system, computer-readable storage media, and the like, capable of storing and retrieving data such as file 22. In some examples, file repository 44 may be part of database 42. Similarly, file repository 44 may be part of server device 38, server device cluster 46, or any combination thereof.

[0033] In addition to transmitting file 22 to client device 4A, document management system 40 updates database 42 to associate an editing lock for file 22 with client device 4A. For instance, document management system 40 may update database 42 to associate a file editing lock with an indication of file 22 (e.g., a file name, a file identification number, etc.) and an indication of client device 4A. As such, document management system 40 may reject editing requests received from client devices other than client device 4A until file 22 is unlocked (e.g., client device 4A checks the file back into document management system 40).

[0034] After receiving file download 26 including file 22 from document management system 40, client device 4A may modify file 22 and store the modified file within local memory of client device 4A. As an example, file 22 may be a text file. A user of client device 4A may modify file 22 using application 6 (e.g., a word processing application). For

instance, the user may modify the text of file 22 and cause application 6 to issue a file save command (e.g., system call 10) to operating system 16. In response, operating system 16 may cause file system 20 to store, modify, or otherwise update file 22 in local memory of client device 4A. In such an example, file system monitor 8 determines that file 22 has been modified and transmits synchronization intent message 28 to document management system 40.

[0035] Document management agent 9 may be any software application executing on client device 4A to operate in conjunction with document management system 40 to synchronize, lock, unlock, or otherwise control access and editing of one or more files associated with document management system 40 (e.g., file 22). Document management agent 9 may be tightly coupled to document management system 40. In some examples, document management agent 9 may be part of an application or suite of tools installed on client device 4A to interact with document management system 40.

[0036] As illustrated in FIG. 1, document management agent 9 may include file system monitor 8. File system monitor 8 may be any software application executing on client device 4A to determine that one or more files (e.g., file 22) is modified using file system 20. File system monitor 8 may be tightly coupled to operating system 16, file system 20, or both. For instance, operating system 16 may expose API 18 to enable file system monitor 8 to register for file modification events. In response to receiving a file modification system call (e.g., a file save, file write, or other file modification system call), operating system 16 raises a file modification event. File system monitor 8 is notified of the file modification event using API 18. As another example, API 18 may include a callback function that executes one or more functions of file system monitor 8 in response to a file modification event.

[0037] In the example of FIG. 1, application 6 issues file modification system call 10 to operating system 16. In response, operating system 16 causes file system 20 to update file 22 in local memory of client device 4A. In addition, operating system 16 raises file modification event 14 that notifies file system monitor 8 using API 18.

[0038] In response to file modification event 14, file system monitor 8 causes a network interface of client device 4A to transmit synchronization intent message 28 to document management system 40. Synchronization intent message 28 notifies document management system 40 that a version of file 22 stored at file repository 44 of server device 38 is no longer the most current version of file 22. As such, synchronization intent message 28 may be considered a file update notification message to notify document management system 40 that a file (e.g., file 22) was modified by client device 4A.

[0039] Synchronization intent message 28 includes an indication of file 22 to enable document management system 40 to identify file 22 as the modified file stored locally at client device 4A. Such an indication may include a file name, a file identification number, a file name and file version number, or other unique identifier of file 22. In some examples, synchronization intent message 28 includes meta-data relating to file 22, such as a time that client device 4A modified file 22, a version number of file 22, an identifier of a user of client device 4A, and the like. In certain examples, synchronization intent message 28 includes the indication of file 22 and meta-data relating to file 22 without including any portion of file 22. That is, in certain examples, synchronization intent message 28 includes the indication of file 22, meta-data relating to file 22, or both, without including file 22 itself or any portion of the modification to file 22.

[0040] Document management system 40 receives synchronization intent message 28 from client device 4A. In response, document management system 40 updates database 42 to associate a storage location of a current version of file 22 with client device 4A. As illustrated in FIG. 1, document management system 40 receives editing request 34 from client device 4B. Similar to editing request 24, editing request 34 includes an indication of file 22 to request access to edit file 22. In addition, editing request 34 may include one or more of an indication of client device 4B, a user of client device 4B, or other similar information. In response to receiving editing request 34, document management system 40 accesses database 42 to identify a storage location of a current version of file 22 and whether file 22 is currently locked for editing by a client device.

[0041] In the example of FIG. 1, document management system 40 accesses database 42 to determine that client device 4A stores the most recently modified version of file 22 (i.e., the current version of file 22) based on synchronization intent message 28 received from client device 4A. In some examples, document management system 40 may determine that file 22 is currently locked for editing by client device 4A (e.g., file 22 is checked out of document management system 40 for editing by client device 4A). In such examples, document management system 40 may reject editing request 34, thereby preventing client device 4B from editing the current version of file 22. In the illustrated example of FIG. 1, document management system 40 determines that client device 4A has relinquished the lock associated with file 22. For instance, at the request of a user, client device 4A may transmit a file unlock message to server device 38 to release an editing lock associated with file 22 and effectively check file 22 back into document management system 40. Rather than transmit file 22 to server device 38 upon releasing the editing lock associated with file 22, client device 4A maintains file 22 locally at client device 4A. In response to determining that file 22 is unlocked, document management system 40 transmits file upload request 30 to client device 4A.

[0042] File upload request 30 includes an indication of file 22 to request that client device 4A transmit file 22 to server device 38. As an example, file upload request 30 may include one or more of a file name of file 22, a file version of file 22, or other information to uniquely identify file 22. In addition, file upload request 30 may include an indication of server device 38 or a server of server device cluster 46 (e.g., a MAC address, IP port number, etc.) to indicate the server device to which client 4A is requested to upload file 22.

[0043] Document management agent 9 executing on client device 4A receives file upload request 30. In response, document management agent 9 transmits file 22 to server device 38. In certain examples, as when file upload request 30 includes an indication of a destination server of server device cluster 46, rather than transmit file 22 to server device 38, document management agent 9 transmits file 22 to the indicated server device of server device cluster 46.

[0044] In the example of FIG. 1, document management system 40 receives file upload 32 including file 22 from client device 4A. Document management system 40 thereafter transmits file 22 to client device 4B. In addition, document management system 40 updates database 42 to associate an editing lock for file 22 with client device 4B. Client device 4B receives file download 36 including file 22 from document management system 40 of server device 38.

[0045] The techniques described above with respect to FIG. 1 may be applied with respect to modifications of file 22 by client device 4B. For example, client device 4B may edit file 22 and store the modified file 22 within local memory of client device 4B. In response to the modification, client device 4B may transmit a synchronization intent message to document management system 40. Such a synchronization intent message from client device 4B notifies document management system 40 that client device 4B modified file 22, and that client device 4B stores the current version of file 22. As such, document management system 40 updates database 42 to associate the storage location of the current version of file 22 with client device 4B. Similarly, client device 4B may transmit a file unlock message to document management system 40 to effectively check file 22 back into document management system 40 while maintaining the current version of file 22 at client device 4B. In response to receiving an editing request for file 22 from a third client device, such as a client device 4C (not illustrated), document management system 40 determines that client device 4B has unlocked file 22, transmits a file upload request to client device 4B, receives file 22 from client 4B, and transmits file 22 to the third client device.

[0046] As such, techniques described herein may reduce the amount of network communications required to manage synchronization and access to a current version of a file associated with a document management system. Rather than transmit an updated file to the server each time the file is modified by a client device, the techniques allow a notification message to be sent to the server device indicating that the client device modified the file. The notification message may be much smaller than the modified file itself. As such, the amount of network communications required to synchronize a current version of a file may be reduced. Moreover, instead of transmitting the updated file to the server device upon completion of the editing process, techniques described herein allow the client device to

store the modified file locally and transmit the file to the server on demand, thereby possibly further reducing the amount of network communications required to manage access and synchronization of a file associated with a document management system.

[0047] FIG. 2 is a flow diagram illustrating an example operation of client device 4A of the communication system of FIG. 1 in further detail. A client device may send a file editing request to a server device to request access to edit a file (50). For instance, document management 9 executing on client device 4A may send file editing request 24 to server device 38 to request access to edit file 22. The file editing request includes an indication of the file (e.g., a file name of file 22, a file version of file 22, etc.) to enable the server device to uniquely identify the file associated with the editing request.

[0048] The server device may receive the file editing request (52). For instance, server device 38 may receive editing request 24 from client device 4A. The server device may obtain and send the requested file (54). As an example, document management system 40 of server device 38 may access database 42 to determine a storage location of a current version of file 22. Server device 38 may obtain the current version of file 22 from the identified storage location (e.g., client device 4B). Thereafter, server device 38 may transmit the current version of file 22 to client device 4A.

[0049] The client device may receive the transmitted file from the server device (56). For instance, client device 4A may receive file 22 from server device 38. The client device may modify the received file locally at the client device (58). For instance, client device 4A may receive file 22 from server device 38. As one example, file 22 may be a text file. Client device 4A may modify the text file locally at client device 4A using application 6 (e.g., a word processing application).

[0050] The client device may determine whether the modified file has been saved within local memory of the client device (60). For instance, application 6 executing on client device 4A may modify file 22. Application 6 may issue one or more system calls 10 to operating system 16 to cause file system 20 to modify, update, or otherwise store the modified file 22 at an underlying physical storage media of client device 4A (e.g., a hard

disk, disc drive, or other computer-readable storage media). Operating system 16 may notify file system monitor 8 of the modification to file 22 using API 18. When file system monitor 8 determines that the modified file has not been saved locally at client device 4A (e.g., file system monitor 8 has not been notified of a file modification event associated with file 22) ("NO" branch of 60), client device 4A may continue to modify the received file.

[0051] When file system monitor 8 determines that the modified file has been saved locally at client device 4A (e.g., operating system 16 notifies file system monitor 8 of a file modification event associated with file 22) ("YES" branch of 60), the client device may determine whether a synchronization intent message has been sent to the server device to notify the server device that the client device has modified the file (62). For instance, document management agent 9 may cause client device 4A to store a flag or other identifier to identify whether a synchronization intent message has previously been sent to the server device in response to a file modification event associated with the modified file. When the client device determines that a synchronization intent message has been sent to the server device ("YES" branch of 62), the client device may refrain from sending an additional synchronization intent message to the server device. As such, the client device may further modify and save the file without triggering additional synchronization intent messages. This may be advantageous to further decrease the amount of network communications associated with synchronization of the modified file among the server device and other client devices.

[0052] When the client device determines that a synchronization intent message has not been sent to the server device ("NO" branch of 62), the client device may send a synchronization intent message to the server device to notify the server device that the client device has modified the file (64). For example, file system monitor 8 executing on client device 4A may send synchronization intent message 28 to server device 38. As such, the synchronization intent message may be considered a file update notification message to notify the server device that the client device has modified the file.

[0053] The server device may receive the synchronization intent message (66), and may record the synchronization (68). For instance, server device 38 may receive synchronization intent message 28 from client device 4A including an indication of file 22. In response,

document management system 40 may update database 42 to associate the synchronization intent with client device 4A.

[0054] In certain examples, the client device may send the updated file to the server device. For instance, the client device may determine that the modified file has been saved locally at the client device, and that the client device has received or generated a system shutdown event. For instance, a user of the client device may provide an input to cause the client device to shutdown (e.g., a user presses a power button of the client device, selects a shutdown option from a user interface of the client device, or otherwise indicates that the client device is to power down). As another example, the client device may receive a system shutdown command from another device, such as an administrator device, to cause the client device to generate a system shutdown event. As such, when in a shutdown state, the client device may not be available to respond to a request to upload the modified file to the server device. In such examples, the client device may send the modified file to the server device in response to the system shutdown event and prior to the client device shutting down. For instance, operating system 16 may notify document management agent 9 that the system shutdown event has been raised. Such notification may be implemented, in certain examples, using an API such as API 18. In response, document management agent 9 may cause client device 4A to transmit file 22 to server device 38 which may store the updated file 22 at file repository 44.

[0055] FIG. 3 is a flow diagram illustrating an example operation of a server device of the communication system of FIG. 1 in further detail. The server device may receive a file editing request message from a client device (70). As an example, server device 38 may receive file editing request 24 from client device 4A to request access to edit file 22. The server device may determine whether the file associated with the file editing request is currently locked for editing by a client device (72). For instance, database 42 may include information to associate one or more files with a file lock identifier. In addition, database 42 may associate an identifier of a client device with the lock identifier. Document management system 40 may access database 42 to determine whether the file is currently locked.

[0056] When the file associated with the received editing request is not locked ("NO" branch of 72), the server device may identify a client device associated with a synchronization intent message associated with the file (74). As such, the server device may identify a storage location of the current (i.e., the most recently modified) version of the file. As an example, document management system 40 may access database 42 to determine an identifier of a client device (e.g., a MAC address, IP address and port number, and the like) to determine the storage location of the current version of the file. Thereafter, the server device may send a file upload request to the identified client device to request the client device to transmit the current version of the file to the server (84).

[0057] When the file associated with the received editing request is locked ("YES" branch of 72), the server device may identify the client device associated with the lock. For instance, database 42 may include information to associate the editing lock of the file with a particular client device. Document management system 40 may access database 42 to identify the client device associated with the lock. The server device may send an unlock request message to the identified client device (78). For example, a user of the identified client device associated with the editing lock may be finished editing the file, or otherwise willing to relinquish the lock. In response to receiving the unlock request, the identified client device may provide an indication to the user, such as through a graphical user interface provided by the client device, requesting the user to unlock the file. For instance, document management 9 may generate a popup window including a graphical textual message and one or more graphical buttons at a display of the client device to enable the user to provide an input in response to the unlock request. In such an example, when the user provides an input to confirm that the client device is to unlock the file, the document management agent 9 may send an unlock message, including an indication of the file, to the server device.

[0058] The server device may determine whether the server device has received an unlock confirmation message associated with the file from the identified client device (80). When the server device does not receive the unlock confirmation message from the identified client device ("NO" branch of 80), the server device may reject the file editing request (82). As an example, the server device may receive an unlock rejection message from the identified

client device indicating that the client device does not relinquish the lock associated with the file. In such an example, the server device may reject the received file editing request. In certain examples, the server device may transmit a file editing rejection message to the requesting client device indicating that the editing request has been rejected.

[0059] As another example, the server device may wait for a predetermined threshold amount of time to receive an unlock confirmation message from the identified client device associated with the lock. When the server device does not receive the unlock confirmation message and the amount of time exceeds the threshold amount of time, the server device may reject the file editing request, and may transmit a file editing rejection message to the requesting client device.

[0060] When the server device receives the unlock confirmation message from the identified client device associated with the lock ("YES" branch of 80), the server device may send a file upload request to the identified client device (84). In addition, the server device may disassociate a file editing lock from an indication of the file and an indication of the client device, thereby unlocking the file for editing. As an example, server device 38 may update database 42 to disassociate a file editing lock from an indication of file 22 and an indication of client device 4A. In addition, server device may send upload request 30, including an indication of file 22, to document management agent 9 executing on client device 4A to request client 4A to transmit file 22 to server device 38.

[0061] As another example, rather than transmit an upload request message to the identified client device, the server device may transmit a message to the identified client device to initiate peer-to-peer communications between the identified client device and the requesting client device. For instance, the server device may transmit network communications information, such as a MAC address, IP address, port number, or combinations thereof, to enable the identified client device and requesting client device to initiate communications over a network. As such, rather than transmit the updated file (e.g., file 22) to the server device (e.g., server device 38), the identified client device (e.g., client device 4A) may transmit the updated file to the requesting client device (e.g., client device 4B) using the peer-to-peer communications.

[0062] In examples when the server device transmits the file upload request to the identified client device, the server device may receive the file the from the client device (86). For instance, server device 38 may receive file upload 32, including one or more portions of file 22, from document management agent 9 executing on client device 4A. The server device may associate an editing lock of the received with the requesting client device (88). For example, document management system 40 of server device 38 may update database 42 to associate the file editing lock of the file with the requesting client device (e.g., client 4B). The server device may send the file to the requesting client device (90). For instance, server device 38 may transmit file download message 36, including one or more portions of file 22, to client device 4B. The techniques described above may be implemented with respect to the requesting client (e.g., client device 4B). As an example, the server device may receive a request to edit the file from a third client device, separate from the identified client device and requesting client device in the example described above. As such, the techniques described above may be applicable to enable the server device to obtain and send the current version of the file to the third client device.

[0063] The techniques described in this disclosure may be implemented, at least in part, in hardware, software, firmware, or any combination thereof. For example, one or more aspects of this disclosure may be implemented using one or more computing devices. Such computing devices may include or be a part of an application server, database server, workstation, or other computing system. Examples of such computing devices may include, but are not limited to, one or more portable computing devices (e.g., a mobile phone, netbook, laptop, personal digital assistant (PDA), tablet device, and the like), one or more desktop computer, or one or more servers.

[0064] Various aspects of the described techniques may be implemented within one or more processors, including one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), or any other equivalent integrated or discrete logic circuitry, as well as any combinations of such components. The term "processor" or "processing circuitry" may generally refer to any of the foregoing logic circuitry, alone or in combination with other logic circuitry, or any

other equivalent circuitry. A control unit including hardware may also perform one or more of the techniques of this disclosure.

[0065] Such hardware, software, and firmware may be implemented within the same device or within separate devices to support the various techniques described in this disclosure. In addition, any of the described units, modules or components may be implemented together or separately as discrete but interoperable logic devices. Depiction of different features as modules or units is intended to highlight different functional aspects and does not necessarily imply that such modules or units must be realized by separate hardware, firmware, or software components. Rather, functionality associated with one or more modules or units may be performed by separate hardware, firmware, or software components, or integrated within common or separate hardware, firmware, or software components.

[0066] The techniques described in this disclosure may also be embodied or encoded in an article of manufacture including a computer-readable storage medium encoded with instructions. Instructions embedded or encoded in an article of manufacture including a computer-readable storage medium encoded, may cause one or more programmable processors, or other processors, to implement one or more of the techniques described herein, such as when instructions included or encoded in the computer-readable storage medium are executed by the one or more processors. Computer readable storage media may include random access memory (RAM), read only memory (ROM), programmable read only memory (PROM), erasable programmable read only memory (EPROM), electronically erasable programmable read only memory (EEPROM), flash memory, a hard disk, a compact disc ROM (CD-ROM), a floppy disk, a cassette, magnetic media, optical media, or other computer readable media. In some examples, an article of manufacture may include one or more computer-readable storage media.

[0067] In some examples, a computer-readable storage medium may include a non-transitory medium. The term "non-transitory" may indicate that the storage medium is not embodied in a carrier wave or a propagated signal. In certain examples, a non-transitory storage medium may store data that can, over time, change (e.g., in RAM or cache).

[0068] As will be appreciated by one skilled in the art, aspects of the present disclosure may be embodied as a system, method or computer program product. Accordingly, aspects of the present disclosure may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system." Furthermore, aspects of the present disclosure may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

[0069] Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

[0070] A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as part of a carrier wave. Such a propagated signal may take any of a variety of forms, including, but not limited to, electro-magnetic, optical, or any suitable combination thereof. A computer readable signal medium may be any computer readable medium that is not a computer readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device. Program code embodied on a computer readable medium may be transmitted using any appropriate

medium, including but not limited to wireless, wireline, optical fiber cable, RF, etc., or any suitable combination of the foregoing. Computer program code for carrying out operations for aspects of the present disclosure may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

[0071] Aspects of the present disclosure are described with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the disclosure. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0072] These computer program instructions may also be stored in a computer readable medium that can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable medium produce an article of manufacture including instructions which implement the function/act specified in the flowchart and/or block diagram block or blocks. The computer program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other devices to cause a series of operational steps to be

performed on the computer, other programmable apparatus or other devices to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide processes for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0073] The flowchart and block diagrams in the Figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments of the present disclosure. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

[0074] Various aspects of the disclosure have been described. These and other aspects are within the scope of the following claims.

CLAIMS

1. A method comprising:

receiving, by a document management system executing on one or more processors of a server device, a file update notification message from a first client device to notify the document management system that a file was modified by the first client device, wherein the file update notification message includes an indication of the modified file on the first client device without including the modified file;

receiving, by the document management system, a file editing request from a second client device to request the file from the document management system;

responsive to receiving the file editing request from the second client device, sending, by the document management system, a file upload request to the first client device to request the modified file from the first client device;

receiving, by the document management system, the modified file from the first client device; and

sending, by the document management system, the modified file to the second client device.

2. The method of claim 1, wherein sending the file upload request to the first client device is responsive to determining, by the document management system and based on the received file update notification message, that the first client device stores a most recently modified version of the file.

3. The method of claim 1, wherein the file update notification message further includes at least one of an indication of a time when the first client device modified the file, a version number of the file, or an identifier of a user of the first client device.

4. The method of claim 1, further comprising:

receiving, by the document management system, a file editing request from the first client device to request the file from the document management system; and

associating, by the document management system, a file editing lock with the indication of the file and an indication of the first client device.

5. The method of claim 4, further comprising:

responsive to receiving the file editing request from the second client device, sending, by the document management system, a file editing unlock request to the first client device;

receiving, by the document management system, an unlock confirmation message from the first client device to indicate that the first client device relinquishes the file editing lock associated with the file; and

responsive to receiving the unlock confirmation message, disassociating the file editing lock from the indication of the file and the indication of the first client device.

6. The method of claim 1, further comprising:

associating, by the document management system, a file editing lock with the indication of the file and an indication of the second client device;

receiving, by the document management system, a file editing request from a third client device;

responsive to receiving the file editing request from the third client device, sending, by the document management system, a file editing unlock request to the second client device;

receiving, by the document management system, an unlock rejection message from the second client device to indicate that the second client device does not relinquish the editing lock associated with the file; and

responsive to receiving the unlock rejection message, sending, by the document management system, a file editing rejection message to the third client device.

7. A method comprising:

modifying a file by a client device;

storing the modified file at a computer-readable storage media of the client device;

sending, by the client device, a file update notification message to a document management system executing on one or more processors of a server device to notify the document management system that the client device modified the file, wherein the file update notification message includes an indication of the modified file stored at the computer-readable storage media of the client device without including the modified file.

8. The method of claim 7, wherein sending the file update notification message to the document management system is responsive to determining, by a file system monitor executing on one or more processors of the client device, that the client device modified the file.

9. The method of claim 7, further comprising:

receiving, by the client device, a file upload request from the document management system to request the modified file from the client device; and

sending, by the client device, the modified file to the document management system.

10. The method of claim 7, wherein the file update notification message further includes at least one of an indication of a time when the client device modified the file, a version number of the file, or an identifier of a user of the client device.

11. The method of claim 7, further comprising:

receiving, by the client device, a file editing unlock request message from the document management system, wherein the file editing unlock request message includes an indication of the modified file; and

sending, by the client device, an unlock rejection message to the document management system to indicate that the client device does not relinquish an editing lock associated with the file.

12. The method of claim 7, further comprising:

receiving, by the client device, a file editing unlock request message from the document management system, wherein the file editing unlock request message includes an indication of the modified file; and

sending, by the client device, an unlock confirmation message to the document management system to indicate that the client device relinquishes an editing lock associated with the file.

13. A server device, comprising:

one or more processors;

a document management system executable by the one or more processors to:

receive a file update notification message from a first client device to notify the document management system that a file was modified by the first client device, wherein the file update notification message includes an indication of the modified file on the first client device without including the modified file;

receive a file editing request from a second client device to request the file from the document management system;

responsive to receiving the file editing request from the second client device, send a file upload request to the first client device to request the modified file from the first client device;

receive the modified file from the first client device; and

send the modified file to the second client device.

14. The server device of claim 13, wherein the document management system is further executable by the one or more processors to send the file upload request to the first client device responsive to determining, based on the received file update notification message, that the first client device stores a most recently modified version of the file.

15. The server device of claim 13, wherein the file update message further includes at least one of an indication of a time when the first client device modified the file, a version number of the file, or an identifier of a user of the first client device.

16. The server device of claim 13, wherein the document management system is further executable by the one or more processors to:

receive a file editing request from the first client device to request the file from the document management system; and

associate a file editing lock with the indication of the file and an indication of the first client device.

17. The server device of claim 16, wherein the document management system is further executable by the one or more processors to:

responsive to receiving the file editing request from the second client device, send a file editing unlock request to the first client device;

receive an unlock confirmation message from the first client device to indicate that the first client device relinquishes the file editing lock associated with the file; and

responsive to receiving the unlock confirmation message, disassociate the file editing lock from the indication of the file and the indication of the first client device.

18. The server device of claim 13, wherein the document management system is further executable by the one or more processors to:

associate a file editing lock with the indication of the file and an indication of the second client device;

receive a file editing request from a third client device;

responsive to receiving the file editing request from the third client device, send a file editing unlock request to the second client device;

receive an unlock rejection message from the second client device to indicate that the second client device does not relinquish the editing lock associated with the file; and

responsive to receiving the unlock rejection message, send a file editing rejection message to the third client device.

19. A client device, comprising:

one or more processors;

one or more applications executable by the one or more processors to modify a file;

a computer-readable storage medium operable to store the modified file; and

a file system monitor executable by the one or more processors to send a file update notification message to a document management system executing on one or more processors of a server device to notify the document management system that the client device modified the file, wherein the file update notification message includes an indication of the modified file stored at the computer-readable storage medium of the client device without including the modified file.

20. The client device of claim 19, wherein the file system monitor is further executable by the one or more processors to send the file update notification message to the document management system responsive to determining that the client device modified the file.

21. The client device of claim 19, further comprising a document management agent executable by the one or more processors to:

receive a file upload request from the document management system to request the modified file from the client device; and

send the modified file to the document management system.

22. The client device of claim 19, wherein the file update notification message further includes at least one of an indication of a time when the client device modified the file, a version number of the file, or an identifier of a user of the client device.

23. The client device of claim 19, further comprising a document management agent executable by the one or more processors to:

receive a file editing unlock request message from the document management system, wherein the file editing unlock request message includes an indication of the modified file; and

send an unlock rejection message to the document management system to indicate that the client device does not relinquish an editing lock associated with the file.

24. The client device of claim 19, further comprising a document management agent executable by the one or more processors to:

receive a file editing unlock request message from the document management system, wherein the file editing unlock request message includes an indication of the modified file; and

send an unlock confirmation message to the document management system to indicate that the client device relinquishes an editing lock associated with the file.

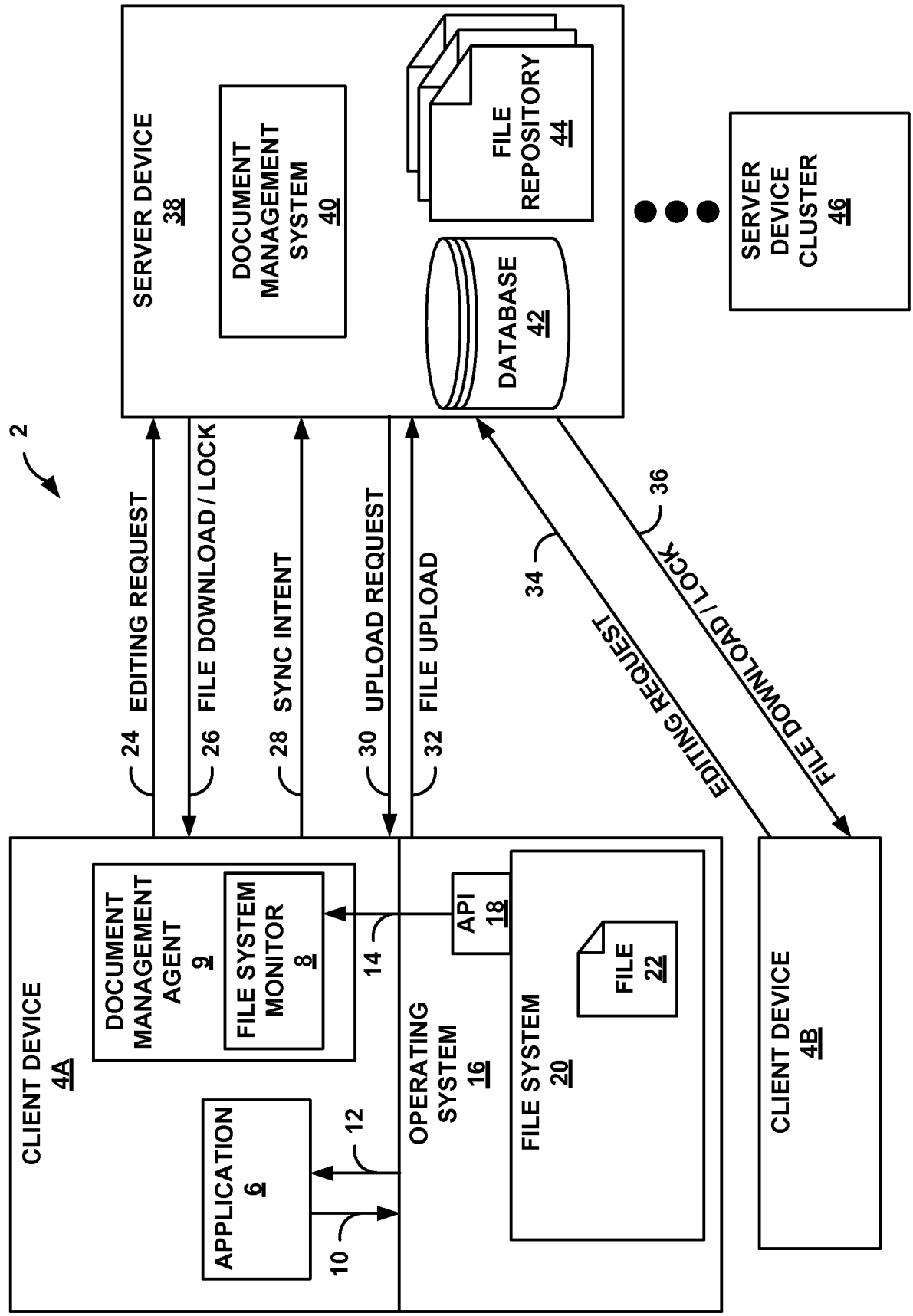
2
SERVER DEVICE
38
DOCUMENT MANAGEMENT SYSTEM
40
FILE REPOSITORY
44
DATABASE
42
SERVER DEVICE CLUSTER
46
24 EDITING REQUEST
26 FILE DOWNLOAD / LOCK
28 SYNC INTENT
30 UPLOAD REQUEST
32 FILE UPLOAD
34 EDITING REQUEST
36 FILE DOWNLOAD / LOCK
CLIENT DEVICE
4A
DOCUMENT MANAGEMENT AGENT
9
FILE SYSTEM MONITOR
8
14
API
18
APPLICATION
6
12
10
OPERATING SYSTEM
16
FILE SYSTEM
20
FILE
22
CLIENT DEVICE
4B

FIG. 1

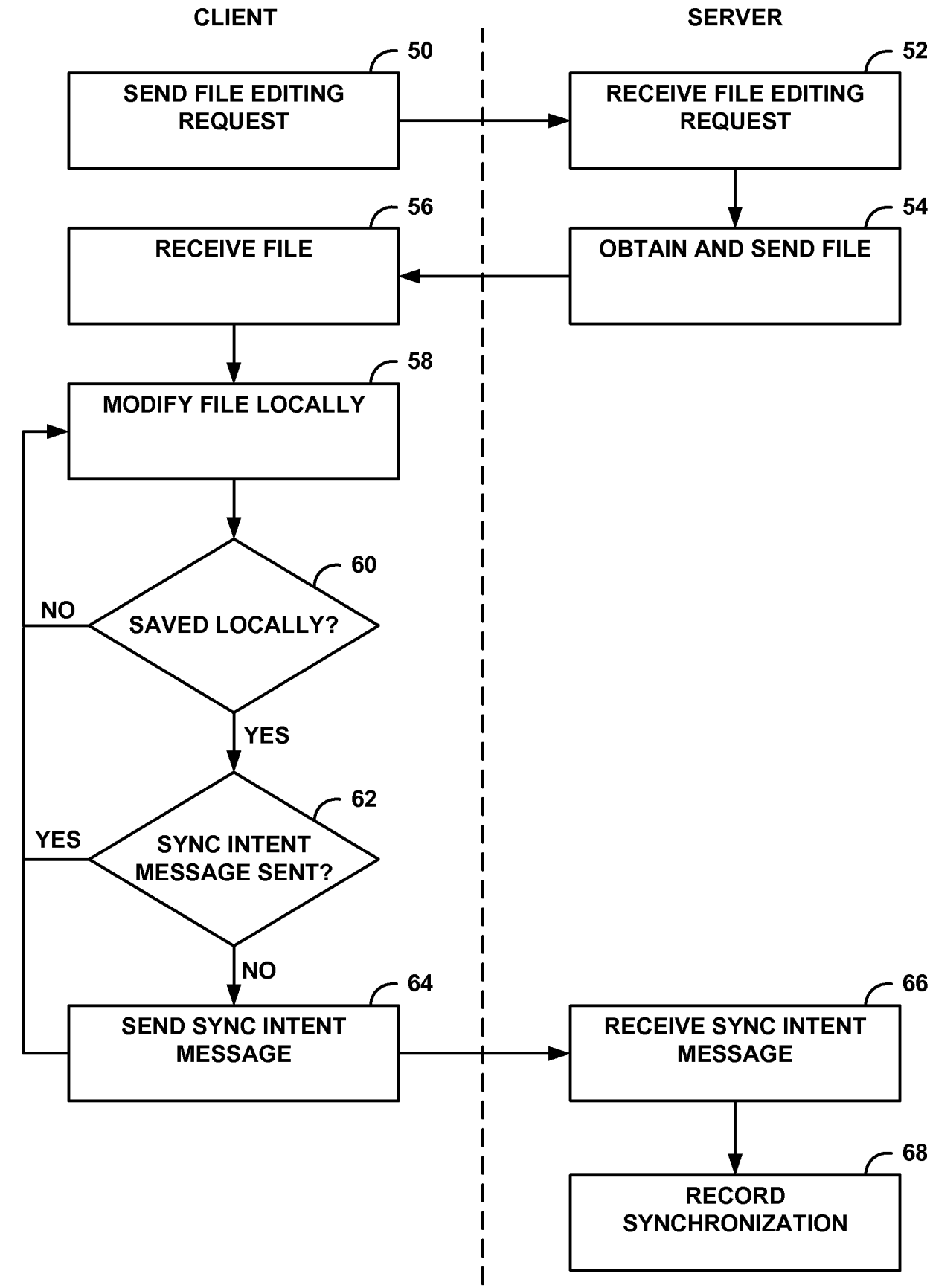
CLIENT
SERVER
50
SEND FILE EDITING REQUEST
52
RECEIVE FILE EDITING REQUEST
56
RECEIVE FILE
54
OBTAIN AND SEND FILE
58
MODIFY FILE LOCALLY
60
SAVED LOCALLY?
NO
YES
62
SYNC INTENT MESSAGE SENT?
YES
NO
64
SEND SYNC INTENT MESSAGE
66
RECEIVE SYNC INTENT MESSAGE
68
RECORD SYNCHRONIZATION

FIG. 2

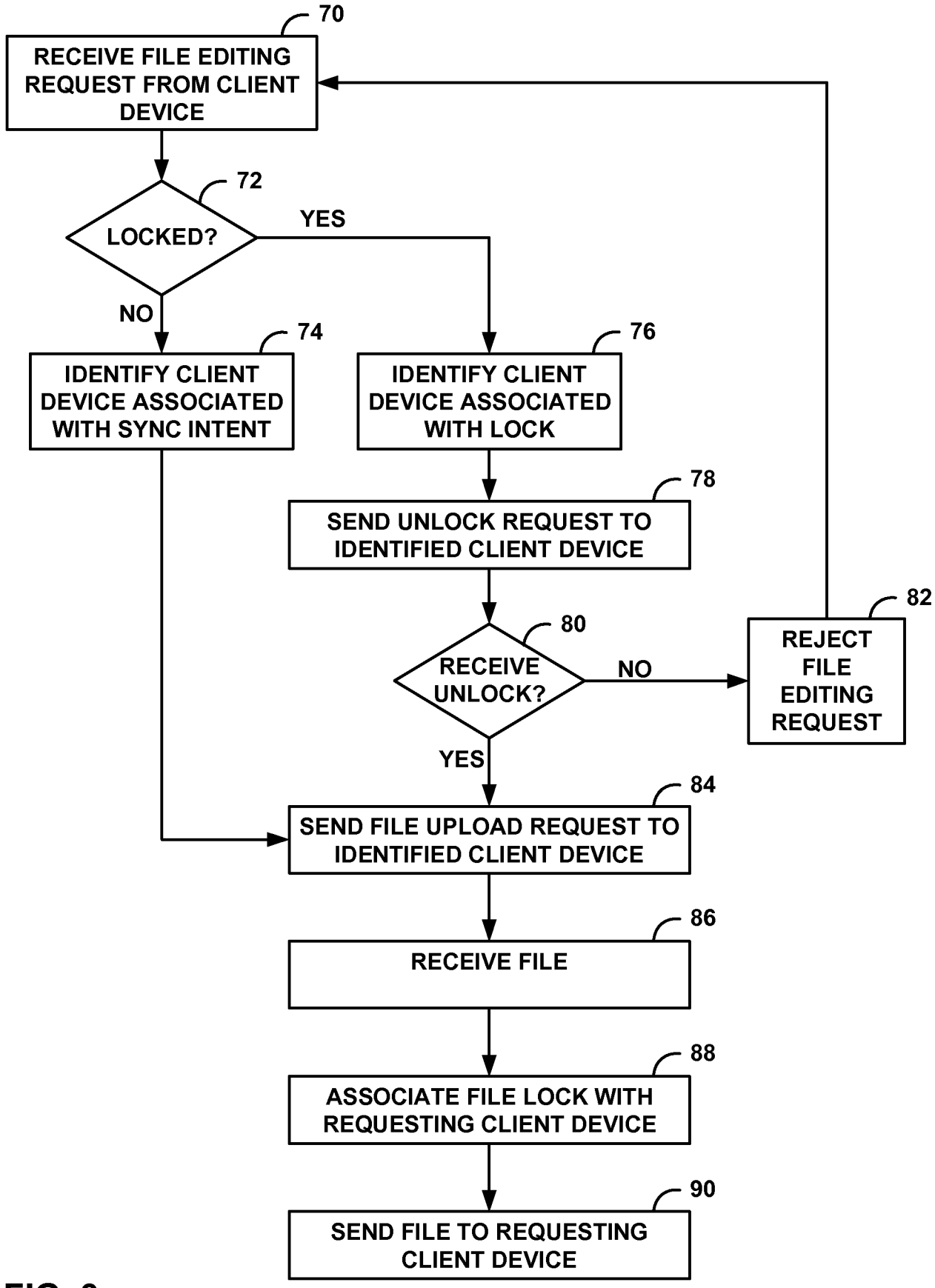
70
RECEIVE FILE EDITING REQUEST FROM CLIENT DEVICE
72
LOCKED?
YES
NO
74
IDENTIFY CLIENT DEVICE ASSOCIATED WITH SYNC INTENT
76
IDENTIFY CLIENT DEVICE ASSOCIATED WITH LOCK
78
SEND UNLOCK REQUEST TO IDENTIFIED CLIENT DEVICE
80
RECEIVE UNLOCK?
NO
82
REJECT FILE EDITING REQUEST
YES
84
SEND FILE UPLOAD REQUEST TO IDENTIFIED CLIENT DEVICE
86
RECEIVE FILE
88
ASSOCIATE FILE LOCK WITH REQUESTING CLIENT DEVICE
90
SEND FILE TO REQUESTING CLIENT DEVICE

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB2013/051343

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. G06F12/00(2006.01)i, G06F17/21(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl. G06F12/00, G06F17/30, G06F17/21

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996
Published unexamined utility model applications of Japan 1971-2013
Registered utility model specifications of Japan 1996-2013
Published registered utility model applications of Japan 1994-2013

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	JP 6-175902 A (FUJI XEROX CO., LTD.) 1994.06.24, [0053]-[0077], Fig. 5-6 (Family: none)	1-3, 7-10, 13-15, 19-22 4-6, 11-12, 16-18, 23-24
Y	WO 2007/089854 A1 (MICROSOFT CORPORATION) 2007.08.09, Page 1, Line 3-14 & JP 2009-525523 A & JP 5009310 B & US 2007/0198657 A1 & EP 1984839 A & KR 10-2008-0100174 A	4-6, 11-12, 16-18, 23-24
A	US 2005/0091448 A1 (HITACHI, LTD.) 2005.04.28, All document & JP 2005-128861 A & US 2007/0271414 A1	1-24

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11.07.2013	23.07.2013

Name and mailing address of the ISA/JP	Authorized officer
Japan Patent Office 3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	Shigeyuki Sakurai Telephone No. +81-3-3581-1101 Ext. 3565

5U 2945

Form PCT/ISA/210 (second sheet) (July 2009)